國立暨南國際大學九十二學年度轉學生入學考試試題

第 3 節程式設計適用:(資工系二 312)

(本試題共 / 頁,第 / 頁)

考生注意: 1. 依次序作答,只要標明題號,不必抄題。
2. 答案必須寫在答案卷上,否則不予計分,並限以藍黑色筆作答。
3. 試題隨卷繳回。(餘請詳閱試場規則)

1. 20%
What are the purposes of ''static'' in following code fragments? (two places)

```
static int f() {}
int g()
{
  static int x;
}
```

2. 80%
Please implement the xstrtok() in C which prototype is:

```
char* xstrtok(char **s, const char* delim);
```

Here is an usage example.

```
char *a = "  ,hello, world, !,  .";  /* the sample source  */
char *d = ", ";                      /* a comma and a blank */
char *s = a;
char *t;
for ( t=xstrtok(&s, d); t != NULL; t=xstrtok(&s, d))
  printf("|%s|\n",t);
```

The output should be: (3 tokens in this example)

```
$ ./a.out      <---- the executable
|hello|        <---
|world|        <---- the output
|!|            <---
$
```

The xstrtok() first skips the characters in ''delim'' (delim is a string, not a single character), then return the token which is consisted of characters with the ones in ''delim'' excluded. For example, in the previous code segment, the first invocation of xstrtok() will first skip the blank and then a comma, and returns ''hello'' as the token. Please note the source string should NOT be altered. That is, you should ALLOCATE space for return value. And after the token returned, you should also update the pointer which is the first parameter. That's why its type is double-indirect.
The detail scenario is:

```
after 1st invocation:  allocate space for "hello" and return it
                       s is updated to point to the comma after "hello".
after 2nd invocation:  allocate space for "world" and return it
                       s is updated to point to the comma after "world".
after 3rd invocation:  allocate space for "!" and return it
                       s is updated to point to the comma after "!".
after 4th invocation:  return NULL (no token found)
                       s is updated to point the last byte of source
                       That is, *s == NULL.
```

NB:

1. You may use malloc() to allocate space. When the allocation fails, just return NULL.
2. DO NOT use any string functions in your code.
   That is, strcpy(), strcat(), strtok() ... all strXXX() are NOT allowed.